Attorney Docket No.: 020375-007400US

PATENT APPLICATION

METHODS AND SYSTEMS FOR EXTRACTING RELATED INFORMATION FROM FLAT FILES

Inventor(s): Larry C. Frame, a citizen of the United States, residing at
7620 North 79th Plaza
Omaha, NE 68122

and

Mark Rowe, a citizen of the United states, residing at
1002 Berkley Ave.
Papilion, NE 68046

Assignee: First Data Corporation
6200 South Quebec Street
Greenwood Village, CO 80111

Entity: Large

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel:

METHODS AND SYSTEMS FOR EXTRACTING RELATED INFORMATION FROM FLAT FILES

BACKGROUND OF THE INVENTION

[01] The present invention relates generally to extracting related data from electronic files. The present invention relates more specifically to systems and methods for selecting related information from one or more two-dimensional data files outside of a relational database environment.

[02] Relational database systems are well known in the art. They are useful for organizing large amounts of data and manipulating and presenting the data in response to "queries," a term well known to those skilled in the art. However, relational database software is often expensive and technically challenging. Further, populating a newly created relational database with preexisting data is often time consuming. Therefore, a primary utility of a relational database system is limited in situations wherein large amounts of data preexist that do not require frequent manipulation.

[03] The need exists for systems and methods for performing relational queries on two-dimensional data files that exist outside of relational database environments. It would be advantageous for such systems and methods to have the capability to relate information from two or more two-dimensional data files.

BRIEF SUMMARY OF THE INVENTION

[04] In one embodiment, the invention provides a method of extracting related information from electronic files. The files each contain a plurality of records for containing data. The method includes comparing data contained in a key segment of each record of a first file to data in a related key segment of each record of a second file. Upon each occurrence of a match of data in the key segment of a record in the first file to data in the related key segment of a record in the second file, a record in a temporary electronic file is created, wherein the record in the temporary file includes data from the records of both the first and second files having matching key segments. Data is selected from records of the temporary file. The data is output. Thereafter, the temporary file may be deleted.

[05] In another embodiment, the method further includes repeating the aforementioned process for additional files, wherein an additional is the first file and the temporary file is the second file.

[06] The first file may be stored in electronic form on magnetic tape. Alternatively the first file may be stored on other media such as, for example, solid state memory, magnetic disk memory, and optical memory.

[07] In another example, the method may also include ordering the records of the first file based on data contained in the at least one key segments.

[08] A record in the temporary file created upon a match of data between records in the first and second files may contain less than all of the data from the matching records of the first and second files.

[09] The particular data from specified records of the temporary file may be selected based in part on logic operators. The logic operators may include less than, greater than, equal to, not-equal-to, less-than-or-equal-to, greater-than-or-equal-to, in and not in.

[10] In another embodiment, the present invention provides a system for extracting related information from electronic files. The system includes a processor, a storage device, and an output device. The system also includes a first electronic file stored on the storage device. The first electronic file contains a plurality of records for organizing data. The system also includes a second electronic file stored on the storage device which contains a plurality of records for organizing data. The processor is configured to compare data contained in a key segment of each record of the first file to data in a related key segment of each record of the second file. The processor is further configured such that upon each occurrence of a match of data in the key segment of a record in the first file to data in the related key segment of a record in the second file, the processor causes a record in a temporary electronic file to be created, wherein the record in the temporary file includes data from the records of both the first and second file having matching key segment data. The processor is further configured to select data from records of the temporary file and output the data to the output device. The processor is further configured to thereafter delete the temporary file.

[11] In another embodiment, the present invention provides a computer-readable medium having computer-executable instructions for performing the method described above. The method includes receiving instructions identifying two or more electronic files from which to extract related information. Each file contains a plurality of records for organizing data. The method also includes receiving instructions identifying a key segment of each record of a first file and a related key segment of each record of a second file. The method further includes

comparing data contained in the key segment of each record of the first file to the related key segment of each record of the second file. Upon each occurrence of a match of data in the key segment of a record in the first file to data in the related key segment of a record in the second file, a record is created in a temporary electronic file, wherein the record in the temporary file includes data from the records of both the first and second file having matching key segment data. Instructions identifying data to be selected from the temporary file are received. The data from records of the temporary file are selected. The data is output. Thereafter the temporary file may be deleted.

[12] In another embodiment, the invention provides a temporary electronic file, comprising at least one record having a plurality of data fields. The data fields are related to data fields of records in a first and a second electronic file. The data in each record of the temporary file are identical to data in the related data fields of either the first or the second file.

BRIEF DESCRIPTION OF THE DRAWINGS

[13] In the figures, similar components and/or features may have the same reference label.

[14] Figure 1 illustrates a system for extracting related information from multiple two-dimensional electronic files according to one example of the present invention.

[15] Figure 2 illustrates a method of extracting related information from multiple two-dimensional electronic files according to one example of the present invention.

[16] Figure 3 illustrates in greater detail an operation of the method of Fig. 2.

[17] Figure 4 illustrates one example of an input data file from which information may be extracted according to the present invention.

[18] Figure 5 illustrates a second example of an input data file from which information may be extracted according to the present invention.

[19] Figure 6 illustrates a temporary data file created according to the present invention.

[20] Figure 7 illustrates the output produced according to one example of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[21] The present invention provides systems and methods for relating information between two or more two-dimensional electronic data files. The present invention is suitable for use with very large data files stored on a wide variety of electronic storage media, including magnetic tape, magnetic disk, solid state memory, optical storage and the like. It provides the ability to compare two or more data files based on one or more key fields, and extract data from the data files in response to the comparison and potentially other conditions. The

present invention accomplishes this relational database-like functionality outside a relational database environment in a novel and heretofore unseen way.

[22] The present invention includes a software application program that configures a programmable computer processor to respond to user commands entered through an input device. One example of a command set and associated syntax is provided in Tables 1 and 2 below. Through the command set, a user identifies two or more data files and the conditions upon which the user desires to relate information from the files. The user also specifies the desired output from the files.

TABLE 1

DVSQL Command Syntax

SELECT {DISTINCT} Alias1.(x1,y1), 'literal', Alias2.(x2,y2), etc.
{**COUNT**(*) or **COUNT(DISTINCT** Alias1.(x1,y1)}
{**MAX**(Alias1.(x1,y1)}
{**MIN**(Alias1.(x1,y1)}
{**SUM**(Alias1.x1,y1)} option to be added later
{**AVG**(Alias1.(x1,y1)} option to be added later
{**EXISTS**}
FROM DDname Alias1, DDname Alias2, etc
INTO DDname (if not provided, default of SQLOUT is assumed)
WHERE condition-1 AND/OR condition-2 etc (no parenthesis allowed)
ORDERED BY (x3,y3)A/D, (x4,y4)A/D, etc. (if A or D not provided, default of Ascending is assumed)

General Notation

Alias* ==> a 1 to 8 character alias used as shorthand notation to reference a file specified in the FROM verb
x* ==> a specified displacement into a record
y* ==> the length of the data at the specified displacement

DDname - DD name of the input/output file to be used. The character following allows for later identification of which fields of which file is being referenced with the A.(displ,length) nominclature.

conditions → Alias1.(x1,y1) operator Alias2.(x2,y2) (valid operators =,<>,<,>,<=,>=,IN,NOT IN)
Alias1.(x1,y1) **BETWEEN** 'literal-1' **AND** 'literal-2'
Alias1.(x1,y1) operator 'literal'
Alias1.(x1,y1) class (ALPHA, INTEGER,ALPHANUMERIC)
Alias1.(x1,y1) **IN** (literal-1,literal-2,literal-3,....,literal-x)
Alias1.(x1,y1) **IN (SELECT** that defines a singular column table of values)
Alias1.(x1,y1) **NOT IN (SELECT** that defines a singular column table of values)

COLUMN FUNCTIONS

DISTINCT	- Eliminate all duplicate keyed output record entries
MAX/MIN	- Output record with the largest/smallest value on the field referenced
COUNT	- Give number of rows in a table (records in a file) or if format COUNT(DISTINCT field), it gives the number of distinct values of the field found in the table/file.
AVG	- Computes the average value of the specified field argument
SUM	- Per specified fields, adds together all the values in that column

TABLE 2

DVSQL EXEC PARM Options

DVSQL is invoked via the SYSTSIN REXX input stream and has execution environment PARMs available to better control the query output/s.

The format is as follows:

```
//SYSTSIN DD *
%DVSQL PNODE {WKDISP} {OUTDSN}
```

PNODE – (mandatory) the value of the primary node that all dynamically generated work data sets used in the DVSQL processing will be cataloged under

WKDISP – (optional) disposition of the work file data sets after DVSQL processing has finished. Valid values are KEEP, *, and DELETE. KEEP will cause all of the work data sets to remain cataloged in the system. * and DELETE (the default) cause all of the DVSQL work data sets to be deleted.

OUTDSN – (optional) data set name to be used to store final DVSQL output in after processing has completed. If the INTO verb is used in the primary level DVSQL, this parameter will be ignored. Also note that if this parameter is used, a value for the WKDISP PARM must also be provided.

[23] In response, the processor creates a temporary file containing the necessary data from the records of the user-specified files. The temporary file may exist only for a brief duration, or may be stored for other uses. The temporary file may exist on solid state memory associated with the processor, or the temporary file may be written to non-volatile memory such as magnetic or optical memory, or the like.

[24] Once the necessary data from the user-specified files are contained in the temporary file, the processor produces output for the user based on the user-specified conditions. The output may be to a permanent electronic file, to a computer monitor, to a printer or the like. Having obtained the necessary information for the temporary file, the processor may then cause the temporary file to be deleted from memory. The following figures and description further explain the present invention.

[25] Fig. 1 illustrates a system 100 according to one embodiment of the present invention. The system includes a processor 102 and a data storage device 104. The processor may be, for example, a mainframe computer, a personal computer, a workstation, or other programmable computing device. The processor is specifically programmed to perform the method according to the present invention. The data storage device 104 may be any one or a combination of, for example, solid state memory such as RAM, a disk drive or series of disk drives, a magnetic tape device, an optical storage system such as a compact disk drive, or

other suitable data storage system. As used herein, data storage device may refer to more than one storage device, and the devices may be located in different physical locations.

[26] The data storage device 104 contains two or more data files each containing one or more records of data. Data files, records and other database terms used herein are familiar to those skilled in the art. Reference is made throughout this description to data or records being in a certain order. While "order" will refer to a particular conceptual arrangement of information, it is not necessarily the case that the information is so arranged on the storage medium of the particular data storage device or devices.

[27] The system 100 also includes an input device 106 through which the processor may be controlled and/or programmed. The input device 106 may also be used to enter data into the records of data files. The input device 106 may be any one or a combination of, for example, a keyboard, a mouse, a microphone or the like. The system 100 also includes an output device 108 for producing a visual representation of the result created by the present invention. The output device 108 may be a printer, computer monitor or the like.

[28] Fig. 2 illustrates a method 200 for selecting related data from two or more electronic data files according to an embodiment of the present invention. The method 200 is performed on the system 100 of Fig. 1 according to the programming of the processor 102. The method 200 begins at operation 202 wherein two data files are compared. As previously stated, the data files are located on the data storage device 104. The operation 202 of comparing the data files is presented in more detail in Fig. 3.

[29] As previously stated, the data files include one or more records of data. The records may be divided into fields of particular data types, although this is not necessarily the case. The records may each contain a sequence of undefined data. However, it is preferable that the records contain similar types of information in the same locations. At step 302 of Fig. 3, a key field or data segment of the record is selected. The key field or data segment contains the data through which the two data files are related. A non-limiting example will be provided hereinafter to more clearly describe this aspect of the present invention.

[30] At step 304, both data files are sorted according to the data contained in the key field or data segment. Step 304 may be accomplished in any of a large number of ways well known to those having skill in the art. Although not essential to the successful operation of the present invention, step 304 increases the efficiency of the next step 306 in terms of processing time required to accomplish step 306.

[31] At step 306, the data contained in the selected field or segment of the records of both data files are compared. Because the records in each file are sorted, it is not necessary to

compare every record of one file with every record of the other file. However, this would be the case if the files are not sorted on the selected field or data segment.

[32] Upon the occurrence of matching data, the process continues at operation 204 of Fig. 2. The transition from operation 202 to operation 204 may take place after the files are fully compared or after each occurrence of matching data, in which case the process would transition from operation 202 to operation 204 repeatedly until the files are fully compared.

[33] At operation 204 of Fig. 2, a record is created in a temporary file each time data in the selected field or segment in a record of one file matches data in the selected field or segment in a record of the other file. The newly created record contains all the information of both records having a matching selected field or segment. In another embodiment of the invention, the newly created record contains only data required by a later operation.

[34] At operation 206, information is selected from the temporary file. The selected information may be data from the selected field or segment that was used to relate the two original files. Alternatively, the selected information may come from other fields or segments of the records of the temporary file. The selected information may be conditional. That is, the information may be selected based on the content of any portion of the records of the temporary file. The use of logic functions to select records is well known. Operation 206 also includes displaying the information or otherwise producing a visual representation of the selected information.

[35] At operation 208, the temporary file is deleted. Although not essential to the proper function of the present invention, deleting the temporary file conserves storage space. Alternatively, the temporary file could be stored for later use.

[36] The present invention is not limited to relating two data files. Following operation 204, additional files may be included, as indicated by operation 210. If an additional file is to be included in the process, the additional file and the temporary file become the two files referred to in operation 202. Thereafter, the process is repeated for each additional file.

[37] The foregoing description describes the operation of the present invention generally. It should be noted that the particular data files, the fields and segments used to related the data files and the information selected from the temporary files is determined by a user. The user provides the information to the processor via the input device or by other means well known to those skilled in the art.

[38] Having described the present invention generally, a more detailed description will be provided by way of a specific, non-limiting example. In this example, a user is attempting to relate information from two electronic files, an activity file and a personal address file. A

textual representation of the activity file is illustrated in Fig. 4, and a textual representation of the address file is shown in Fig. 5. The activity file 400 includes an activity field 402, a last name field 404, and a first name field 406. The activity file 400 might represent a listing of people participating in various activities in a sports organization, for example. As can be appreciated, the activity file 400 might list a person with the same first and last names participating in two different activities. For example, "Betty Ames" appears next to entries for both "Aerobics" 408, "Softball" 409, and "Swimming" 410. The address file 500 includes a last name field 502, a first name field 504, a street address field 506, a city field 508, a zip code field 510, and a phone number field 512.

[39] In this example, the user is attempting to determine all the people from the address file who live in the 68134 zip code and play softball in the organization. Thus, the user is attempting to obtain related information from two, two-dimensional data files.

[40] According to the present invention, the user identifies one or more fields of the files to compare. The user may communicate this information to a computer processor using an input device and a command language which the processor understands. The use of command languages are well known to those having skill in the art. As stated previously, Tables 1 and 2 include an example of the syntax of a command language according to an embodiment of the present invention.

[41] In this example, the first name 406, 504 and last name fields 404, 502 are used to relate the two files. Thus, according to the invention, the files are individually sorted using the related fields. A sorted version of the activity file 400 is not shown; however, it is apparent to one skilled in the art how the sorted data would be arranged. The address file 500 is shown in Fig. 5 sorted by last name. As stated previously, the sorting operation is optional, although it increases the efficiency of the method of the present invention.

[42] According to the invention, the files are compared to determine which records of the two files have matching data in the related fields. A temporary file is created having a record for each match in the comparison between the files. Fig. 6 illustrates a temporary file 600 for this example.

[43] As can be appreciated with reference to Fig. 6, the temporary file 600 includes all the fields from both of the two original files 400, 500. However, the temporary file 600 does not necessarily include all the data from the two original files 400, 500. Further, it is also apparent that the last name 404, 502 and first name 406, 504 fields appear twice in the temporary file 600. In another example of the invention, both the related fields might not be included in the temporary file 600 since the fields contain matching data for each record

included in the temporary file 600, according to the present invention. It is further apparent that similar data might be included in the temporary file more than once (e.g., Tina K Hayes 602, 604, 606). However, the appearance of such names in different records is a result of the name fields being associated with different activities in the activity field 402.

[44] It should also be noted that some apparent matches are not included among the records of the temporary file. For example, the address file 500 included "L Petersen" 514 and "Laurence Petersen" appeared several times (412, 414) in the activity file 400. However, "L" does not match "Laurence", according to this example of the present invention; therefore, associated records are not included in the temporary file 600. Other embodiments of the present invention could include logic that either includes or further evaluates such situations.

[45] In an alternative embodiment of the present invention, the content of the temporary file 600 may be reduced in one or more of several ways. First, the method of the present invention might begin by creating an interim temporary file for each of the original files having only data meeting specified criteria. For instance, in this example, the interim file associated with the activity file 400 would include only the data from the records having "softball" in the activity field 402. Also, the interim file associated with the address file 500 would only include the data from the records having "68134" in the zip code field 510. Thus, the size of the temporary file 600 would be substantially reduced.

[46] An additional approach to potentially reducing the size of the temporary file 600 is to limit the data that is entered into the records of the temporary file. For instance, in the present example, the data in the street address field 506 is not used to produce the user-requested output. Therefore, the temporary file 600 might not include the street address field 506, in which case, the size of the temporary file 600 would be further reduced. Other such space-saving measures are now apparent to those skilled in the art in light of this disclosure.

[47] Continuing with this example of the present invention, including the temporary file 600 of Fig. 6, another step in the method will be discussed. The temporary file 600 may now be evaluated to determine which records include data that matches the specified conditions of the present example. Thus, for each record that includes "softball" in the activity field 402 and "68134" in the zip code field 510, the data from the first name 406, 504 and last name 404, 502 fields are provided to the user. This may be accomplished by creating an output record, a printed page, a visual representation on a computer monitor, or the like. One example of such output is illustrated in Fig. 7, which includes the three names from this example that match the specified conditions.

[48] The foregoing discussion illustrates but one example of the present invention. Numerous additional embodiments and equivalents are apparent to those skilled in the art in light of this disclosure. Therefore, the invention is not limited by this disclosure but is intended to be interpreted in light of the following claims.